Tracking GDPR Compliance in Cloud-based Service Delivery

Masoud Barati and Omer Rana

Abstract—The European General Data Protection Regulation (GDPR) has had a far-reaching impact on data privacy and compliance for cloud providers. GDPR influences access to, storage, processing and transmission of personal data, requiring these operations to be verified by a cloud user through explicit consent prior to execution. GDPR rules implemented for such operations can be ambiguous and often open to interpretation, making manual verification a time consuming and error prone process for cloud providers. An encoding of GDPR rules is described, with each operation carried out using these rules recorded into a Blockchain for auditing purposes. Specifically, this work shows how some GDPR rules can appear as *opcodes* in smart contracts to verify the operations of providers on user data in a transparent and automatic way. An abstract model is designed to demonstrate how cloud providers can access and deploy such smart contracts through a Blockchain-based virtual machine. A case study is used to demonstrate how this approach can be used in practice. The case study uses a collection of design patterns and smart contracts to verify provider operations, including *read*, *write*, *execution* and *transfer* on user data. Validation is undertaken by deploying the smart contracts in a Blockchain test network to investigate the execution costs of GDPR compliance checking.

Index Terms—user privacy, cloud security, blockchain, smart contracts, general data protection regulation

1 Introduction

A web service accessed by a user may utilize a number of additional externally hosted services across multiple data centres and providers [14]. Although the user is connected via a web browser to a single end point (web service), there may be a number of additional services that come together to offer this capability – often unknown to the user. Such providers can also increasingly be mobile services at the network edge. Users of these services transfer their personal data through a web browser interface without realizing that the data may be shared among several back-end services. Even where this data is encrypted during transfer, the first hop web server needs to decrypt this. By increasing the number of mobile services and edge resources, a large proportion of cloud users are voicing misgivings with respect to how privacy is handled across this cloud service ecosystem, resulting in limited or no visibility of who has access to data. The General Data Protection Regulation (GDPR) was introduced to ensure that non-expert users can make informed decisions about their privacy and give *informed consent* for the use, sharing, and re-purposing of their personal data [15].

GDPR has been introduced to protect personal data of EU citizens from privacy breaches. The main elements of GDPR are: data subject, data controller or joint controller and data processor. Data subject is directly or indirectly identified through name, location or IP address. Data controller is a person or organization specifying the purposes of data processing. Data joint controller can be defined where two or more controllers jointly determine the purpose of data processing. Data processor generally acts on the behalf of a controller or joint controller [16]. Given theses elements, GDPR delegates the responsibility of any violation in data processing to controller or joint controller, but also gives a shared responsibility to the processor when a user has no direct control on the processing steps involved. For instance, according to GDPR requirements, an infrastructure-as-a-service (IaaS) provider that supplies user with a managed hosting service will have the responsibility of processing data produced by its infrastructure (e.g. the recording and management of system and access logs) [15].

Blockchain technology has also been applied in a cloud environment to enhance user privacy and to provide an audit trail of providers through a distributed, consensus-based approach [18]. Cloud applications can make use of Blockchain-based techniques to enable a user to have control over their data, and be informed about the types of data processing that has been carried out by providers [22]. The integration of GDPR and Blockchain technology enables accountability and provenance tracking of operations carried out on user data [17]. Such an approach relies on the use of auditable smart contracts deployed in a Blockchain, improving the transparency of personal data usage and processing. A conceptual architecture was proposed for a privacy-aware cloud ecosystem that takes advantage of both GDPR and Blockchain [23].

Although GDPR and Blockchain-based techniques have been widely used to support immutable transactions on cloud providers, the verification of GDPR rules relating to operations carried out by cloud providers over personal data is still performed manually. Furthermore, the potential of translating GDPR rules into smart contracts and the automatic detection of violations (and the actors involved) have received limited attention. To address these limitations, this paper provides the following contributions: (i) GDPR rules as *opcodes* in

- *M. Barati is with the School of Computer Science & Informatics, Cardiff University, Cardiff, UK. E-mail: Baratim@cardiff.ac.uk*
- *O. Rana. is with the School of Computer Science & Informatics, Cardiff University, Cardiff, UK. E-mail: ranaof@cardiff.ac.uk*

Manuscript received –, –; revised –.

smart contracts to automatically verify the operations of providers on user data; (ii) defining operations carried out by providers on personal data during the life cycle of a service; (iii) an abstract model description to show how providers can connect to a Blockchain-based virtual machine; (iv) an algorithm for identifying actors executing non-compliant operations (according to GDPR rules) and violating user consent obligations; (v) automatic verification of GDPR compliance using a case study; (vi) experiments demonstrating the potential scalability of the proposed approach, using the *gas*-based metric to identify the computational cost of undertaken GDPR compliance checking.

The remainder of the paper is structured as follows: Section 2 presents background and context about Blockchain and the translation of GDPR rules into pseudo-code. Section 3 proposes a classification scheme that supports the implementation of GDPR rules as smart contracts and formally defines the types of operations carried out on user data by providers, followed by Section 4 describing how smart contracts can be used for verifying GDPR compliance. An architecture is proposed in which such contracts are accessible to both users and providers, followed by a description of components that are used in such an architecture in Section 5. Section 6 presents a case study expressed as a collection of business processes – represented using the Business Process Modelling Notation (BPMN). We annotate the BPMN diagram to describe GDPR requirements associated with activities and data products involved in these processes. Section 7 describes experimental results of our Blockchain-based approach, particularly focusing on the cost of verifying GDPR compliance. Related work is described in Section 8, with conclusion and future work provided in Section 9.

2 Background & Context

This section (briefly) describes Blockchain and smart contract technologies and reviews the translation of GDPR rules into pseudo-code that can be implemented through smart contracts.

2.1 Blockchain

Blockchain is a decentralized ledger storing a set of records in blocks, structured as a linked list. Each block uses the hash address of its previous block, can record a number of transactions and a time stamp showing the creation time of the block. Users interacting with a Blockchain can access the blocks, but they cannot change or delete content in the block. Blockchain contains a network of peer-to-peer nodes called miners, which can add additional blocks based on a consensus algorithm. Mining is a key concept of Blockchain through which a block is created and attached to the Blockchain network [23]. Popular consensus algorithms proposed for mining are: Proof of Work (PoW), Proof of Stake (PoS), Proof of Space (PoSpace), Proof of Importance (PoI), Practical Byzantine Fault Tolerance (PBFT) and Measure of Trust (MoT) – often collectively referred to as Proof of X (PoX).

Blockchain are categorized into public, federated, or private [1]. In a public Blockchain, everyone can participate and access blocks without permission (e.g. Bitcoin [2] and Ethereum [12]). A federated Blockchain is operated under the authority/ engagement of several organizations or groups, which do not permit any user with access to the Internet to take part in the verification of transactions (e.g. Corda [3] and R3 [4]). Finally in a private Blockchain, only one organization has permission for creating or verifying blocks (e.g. Monax [5] and Multichain [6]).

The evolution of Blockchain technology is generally classified across three generations. The first generation is restricted to cryptocurrency transactions implemented via Bitcoin – often identified as the first application using a Blockchain. The second generation enabled users to exchange various types of assets, ranging from goods to (even) votes. The third generation introduces "smart contracts" that can be deployed on the Blockchain network and be checked by users connected to the network [9]. Smart contracts expand the capabilities of a Blockchain and enable the integration of Blockchain techniques into numerous industrial applications.

Smart contracts transform business rules to software code that can be automatically executed on a Blockchain. The execution of smart contract is independent of any third party and the code/script of a contract is recorded in a Blockchain. A smart contract has a set of transactions, each of which may change the state of the Blockchain – such as Ethereum [12]. Developers can use JavaScript, Python, or the Solidity programming language [31] to create a contract in Ethereum. This platform also requires payments in the form of *gas* for deploying a smart contract, or for executing transactions that change Blockchain state. Gas is a unit measuring the amount of computational effort required to execute particular operations in smart contracts, i.e. it is an internal currency in Ethereum to pay for transaction fees. Gas is paid in *ether* – a cryptocurrency in Ethereum that allows smart contracts to be executed. Although the amount of gas used for activating a transaction may be high, its translation to ether unit is very low. For instance, if the consumed gas of a transaction is 10000, the transaction fee is approx. 0.0002 (ETH) [7]. Ether motivates miners to validate blocks in the Ethereum Blockchain, as the successful miner is awarded *ether* units for each validation. Additional details about gas consumption of each operation (opcode) can be found in [8].

2.2 Translating GDPR Rules into Pseudo-Code

The idea of verifying GDPR compliance in an automated way comes from an approach presented in [10], where the authors identified a number of questions related to legal concerns around GDPR. These questions relate to data protection and privacy measures that should be supported by cloud services. A number of such questions are presented in Fig. 1.

Legal question L1: relates to the sensitivity of user data. In the GDPR standard, sensitive data consists of information such as religious or political beliefs, genetic data, biometric data and health-related data.

Legal question L2: checks if cloud services have a user authentication mechanism, such as secure login for preventing unauthorized access to user data.

Legal question L3: verifies the geographical location of a provider receiving user data (e.g. whether it is in Europe or not).

Legal question L4: checks for Binding Corporate Rules (BCR) certification of non-European data receivers. The BCR is a code of conduct adopted by a community of multinational

- **L1**: Does your service deal with sensitive personal data?
- **L2**: Does your service support encryption or authentication access for the customer data?
- **L3**: Does your service give the choice of EU-based migration of personal data?
- **L4**: Has your underlying connected provider been certified for their Binding Corporate Rules (BCR) clauses by a EU DPA?

...

Fig. 1. GDPR legal questions.

```
compliance = true;
if L1 == Yes then
  if L2 == No then
    compliance = false;
    return compliance;
if L3 == No then
  if L4 == No then
    compliance = false;
    return compliance;
```

Fig. 2. Pseudo-code for checking legal compliance.

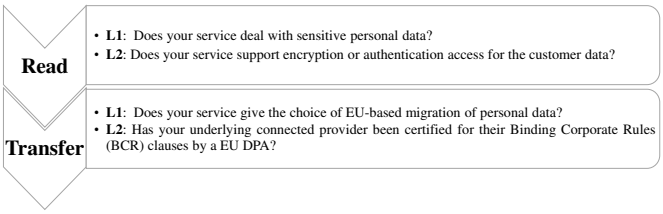

Fig. 3. Two typical operations with their legal questions.

companies that want to move user data internationally across various jurisdictions [10].

To verify provider compliance with GDPR rules as identified above, pseudo-code was created to support automated verification [10]. The pseudo-code first categorizes a legal question into the most relevant GDPR rule(s), followed by a ("`Yes`" or "`No`") confirmation for each question. Example of pseudo-code is provided in Fig. 2. For instance, as legal questions L1 and L2 relate to Art. 32(1)(a) of GDPR, their verification is interdependent (see Fig. 2). The rule states that if personal data is sensitive, the service offered to a customer needs to support encryption or user authentication. L3 and L4 are also considered as related legal questions due to Art. 44–47 of GDPR restricting the transfer of personal data only to countries following BCR rules. Smart contracts can benefit from the pseudo-code description of such rules to translate legal GDPR compliance rules into operations that can be verified using monitoring of cloud provider operations.

3 Data Processing and GDPR

A key assumption behind our approach is that activities carried out by providers (actors) on user data can be classified into a set of operations that can be monitored, e.g. read, write, transfer, etc carried out by one or more cloud providers. For each type of operation, a number of GDPR rules can be proposed. Subsequently, legal questions such as those in Fig. 1 can be assigned to each operation. Hence each rule can be associated with a legal clause in GDPR, and represent a legal question associated with processing of user data. Figure 3 shows an example of such a classification with two operations. The operation *Read* is associated with two legal questions L1 and L2 which are related to the GDPR rule for accessing user data – Art. 32(1)(a). Similarly, operation *Transfer* can involve L3 and L4 and focus on the migration of user data outside Europe – Art. 44–47.

This classification allows the automatic verification of GDPR rules on actors, by providing a more structured view for developers to work with. By assigning available GDPR rules to operations in the form of legal questions, checking of GDPR compliance for operations in an automatic way is provided. An operation can have a number of elements – called as *GDPR-concern* elements – each of which refers to a keyword in a GDPR legal question. For instance, the keyword of the legal question L2 proposed for *Read* operation is data encryption.

It is assumed that a function can be implemented programatically for each operation. The *GDPR-concern* elements are parameters verified on an operation, and used as a basis for checking GDPR compliance. The values are provided by actors and can have binary, textual, or numerical forms. As an example, the country name of an actor receiving personal data can be a parameter used to check compliance of the *Transfer* operation. As shown in Fig. 2, the if-then clause is used to verify whether the operation of an actor complies with GDPR, e.g. if the country of the actor receiving user data is outside Europe, then the transfer of data is likely to be non-compliant.

Definition 1. Let α be an operation and $El = \{el_1, \cdots, el_n\}$ be a set of *GDPR-concern* elements of α such that $el_i \in El$ refers to a keyword or element related to a legal question proposed for α. The following Boolean-valued function is defined for the operation α to show its GDPR compliance status:

$$G_\alpha : \times_{i=1}^{n} V_i \mapsto \{\top, \bot\},$$

where $V_i = dom(el_i)$ is the set of values associated with the domain of el_i. The operation α is GDPR compliant, if $G_\alpha(v_1, \cdots, v_n) = \top$, where $v_i \in V_i$.

3.1 Data Usage Model

Service provision may involve a number of actors (providers) executing a sequence of operations on user data. Operations may collect or use personal data for realizing the service – the actual reasons for data collection/ usage is dependent on the actors involved. We can define a data usage model to formally represent activities of actors during the life cycle of a service.

Definition 2. The data usage model of a service is a tuple: $\mathcal{M} = \langle \mathcal{ACT}, \mathcal{A}, \mathcal{D}, \mathcal{P} \rangle$, where $\mathcal{ACT}$ is a set of actors; $\mathcal{A}$ is a set of operations on user data. The set $\mathcal{D}$ contains data classes that are relevant for a user; $\mathcal{P} \subseteq \mathcal{ACT} \times \mathcal{A} \times \mathcal{D}$ is a data usage relation set, where each relation determines the data used by an actor and the operation to be executed on the data.

The set $\mathcal{D}$ does not refer to specific data values, it only specifies the kind of data involved, e.g. user name and address.

GDPR enforces actors to explicitly define their purpose of data processing, i.e. what operation is carried out by which actor on what data. Moreover, it encourages actors to inform users about the GDPR compliance status of their operations in advance, as defined in GDPR Art. 5 and Art. 30(1)(b). The following definition formally expresses the purpose of data processing.

Definition 3. Let $\mathcal{M} = \langle \mathcal{ACT}, \mathcal{A}, \mathcal{D}, \mathcal{P} \rangle$ express a data usage model. The purpose of data processing is defined by

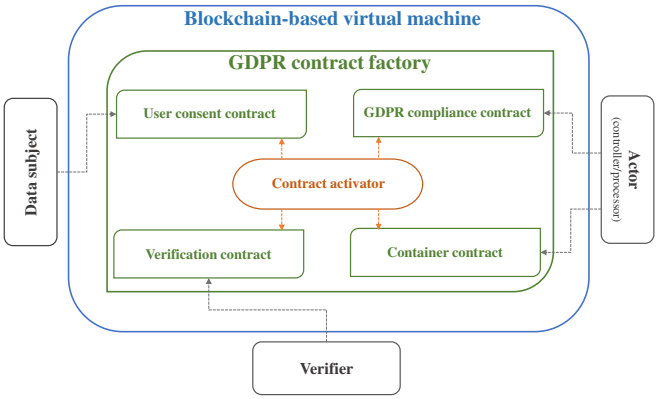

Fig. 4. A GDPR-based architecture for data privacy using Blockchain.

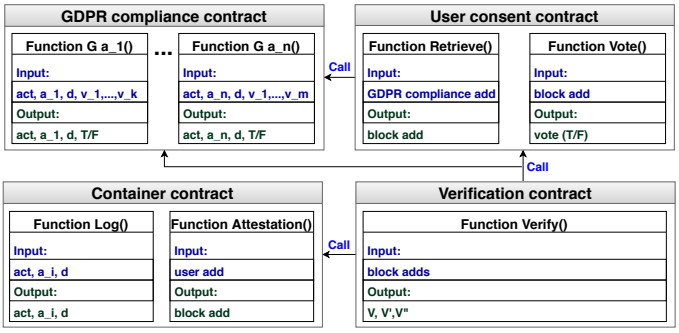

Fig. 5. Abstract model of smart contracts.

a relation set P, referred to as *purpose* relation set, such that $P \subseteq \mathcal{ACT} \times \mathcal{A} \times \mathcal{D} \times \{\top, \bot\}$, where the set $\{\top, \bot\}$ denotes the GDPR compliance status of operations. Each relation specifies the type of operation to be executed by an actor, the data on which this operation is performed, and the GDPR compliance status of the operation.

Assuming that an operation $\alpha \in \mathcal{A}$ of an actor $act \in \mathcal{ACT}$ has the set of GDPR-concern elements $El = \{el_1, \cdots, el_n\}$ and $G_\alpha(v_1, \cdots, v_n) = \top$, where $v_i \in dom(el_i)$. The purpose relation $\langle act, \alpha, d, \top \rangle \in P$ states that the actor will execute α on user data $d \subseteq \mathcal{D}$ and that α is GDPR compliant. Hence, such a relation provides a more transparent mechanism for verifying actor behaviour and their reasons for collecting/ processing personal user data.

4 A Blockchain-based Architecture

Figure 4 shows an architecture for supporting GDPR compliance checking in a cloud system. It enables the development of an audit trail of actors (with the roles of controllers or processors) by using a Blockchain that records the operations executed by actors on user data. Moreover, the architecture can be used to verify whether the executed operations comply with GDPR rules (or not). The main components of the architecture are: a Blockchain-based virtual machine, a GDPR contract factory and a contract activator. The entities interacting with the components are: data subject, actors and verifier.

4.1 Blockchain-based Virtual Machine

A Blockchain platform such as an Ethereum Virtual Machine (EVM) [12] provides an environment for actors to deploy smart contracts and build a Blockchain. The virtual machine also enables compilation, execution and debugging of smart contracts.

4.2 GDPR Contract Factory

The GDPR Contract Factory contains four smart contracts that provide the basis for the verification of actor operations with respect to a set of GDPR rules. The smart contracts are: GDPR compliance contract, user consent contract, container contract, and verification contract. Figure 5 shows an architectural overview of the smart contracts.

GDPR compliance contract captures information required by a data subject before actors can access personal data. The contract implements a number of functions, enabling actors to specify their purpose(s) of data processing. Each function, implemented for a specific operation (i.e. $G_\alpha(v_1, \cdots, v_n)$), processes its inputs (i.e. the values of GDPR-concern elements) based on a set of terms proposed for verifying GDPR rules and determines the GDPR compliance status of the operation (for more details see Section 6).[1] Providing a compliance status can help a user make a more informed decision about accepting or rejecting a data processing request. Based on Def. 3, the contract also requires an actor to submit information about the 'purpose relation' set P into the Blockchain. Each submitted record, specifying the purpose of data processing, contains: (i) the address of actor $act \in \mathcal{ACT}$, (ii) the operation $\alpha \in \mathcal{A}$ that will be executed by the actor, (iii) the personal data $d \subseteq \mathcal{D}$ required for the operation, (iv) the GDPR compliance status of the operation ($\top$ or $\bot$). The contract meets the Art. 30(1)(b) of GDPR under which the purpose of data processing and the address of an actor must be determined in advance. Note that the actor's address can be a Blockchain wallet ID (e.g. an Ethereum account).

User consent contract contains two functions: Retrieve and Vote. The *Retrieve* function uses the address of a GDPR compliance contract to provide a data subject information recorded by an actor in the Blockchain, namely: actor ID, next operation, required data, and GDPR compliance status of the operation. The *Vote* function in Fig. 5 submits a data subject's votes (positive/negative consent) to the Blockchain. This contract realizes Art. 6(1)(a) of GDPR, whereby a data subject must give consent for processing of their personal data. Formally, the following function defines the vote of a data subject for accepting or rejecting the purpose of data processing.

Definition 4. Let $P \subseteq \mathcal{ACT} \times \mathcal{A} \times \mathcal{D} \times \{\top, \bot\}$ be the purpose relation set. The vote (positive or negative consent) of data subject j for the purpose of data processing is denoted by a Boolean function: $\Gamma_j : P \mapsto \{\top, \bot\}$.

Given a relation set $\gamma = \langle act, \alpha, d, \top \rangle \in P$, a consent has been given by the data subject if $\Gamma_j(\gamma) = \top$.

Container contract is used for recording all operations performed by actors (service providers) on personal data

1. The functions in the GDPR compliance contract are activated by actors requesting personal user data. The activation of these functions is recorded in the Blockchain.

within an application container. The audit trail associated with these operations is submitted to a Blockchain. The assumption is that a *trustable* container is launched on the cloud provider to track these operations on personal data. A trustable container has the following properties: (i) it can record operations carried out on data hosted in the container using a monitoring tool, (ii) the log generated by the monitoring tool cannot be modified by the cloud provider on which the container is hosted – as described in the VMInformant system [24]. Trustable containers can be realised in a number of ways – e.g. using Virtual Trusted Platform Modules (vTPM) [25] and Intel Software Guard Extensions (SGX) [26] that protect containers from an untrusted host. These approaches make use of trusted hardware to protect containers from side channel attacks initiated by the hosting platform. Trustable containers can also utilise inter-container protection mechanisms to improve security, e.g. the approach proposed in [27] protects containers from other malicious containers on the same host. Co-hosting a malicious Virtual Machine (VM) on the same physical host as the intended target VM has been a traditional approach adopted in cloud systems in the past. In [28], a container-specific security profile is presented, enabling operations on each container to be managed separately. Encryption of containers has also been undertaken in Docker, where a password, Secure Shell (SSH) private key, Secure Sockets Layer (SSL) certificate etc is used to coordinate data access, limiting what can be visible within particular containers. Using this approach, encrypted data can be transferred across the network, and only containers that have the password (referred to as the "secret") are able to see the data. Another use case for using secrets is to provide a layer of abstraction between the container and a set of credentials – e.g. when separating credentials for production vs. test environments. Kubernetes, another popular container management framework, also supports encryption of data when stored within a container. In general therefore, a trustable container in the context of this work is one which: (i) is able to record operations carried out on data hosted within the container – using, for instance, introspection techniques as reported in VMInformant [24], and which does not require monitoring from the hosting platform; (ii) supports encryption of data hosted within the container.

The Container contract has a function, called *Log*, that is activated by the container. This function collects information about operations carried out in the container, thereby supporting a subset of the data usage relation set $\mathcal{P}' \subseteq \mathcal{P}$ described in Def. 2, and submits it into the Blockchain. Such information includes the address of the actor (cloud provider) *act* involved, the executed operation α, and the data d that has been processed. The Container contract also provides an *Attestation* function which informs the data subject where their personal data is being processed. By running this function a data subject can be informed about the history of data movement between different actors involved in a service provision. The presence of such function meets Art. 15(2) and 20(2) of GDPR.

Verification contract uses the *Verify* function to confirm whether user consent has been obtained. Verification is carried out using transaction log in a Blockchain.

Definition 5. Let $\mathcal{P}' \subseteq \mathcal{P}$ be a subset of data usage relation set formed through the Container contract and $P' = \{\langle act, \alpha, d\rangle \mid \langle act, \alpha, d, \_\rangle \in P \wedge \Gamma_j(\langle act, \alpha, d, \_\rangle) = \top\}$ be a relation set derived from the records submitted by the GDPR compliance contract and which has received user consent. A violation in Art. 6(1)(a) of GDPR can be flagged (i.e. the rule legislated for data subject consent) if $\mathcal{P}' \not\subseteq P'$.

Given this definition, the *Verify* function detects possible violations through Algorithm 1. Actors who violate GDPR rules are classified into three groups: high-risk, medium-risk, and low-risk. High-risk actors are denoted by $\mathcal{V}$ in Algorithm 1, i.e. operations are not GDPR compliant and have not received user consent. Medium-risk actors are denoted by $\mathcal{V}'$ and execute GDPR compliant operations, but the executions have not been accepted by a data subject. Low-risk actors are denoted by $\mathcal{V}''$ and refer to those who have obtained user consent but their operations are not GDPR compliant.

Algorithm 1 The verification of actors

Let $\mathcal{V}$ be a set of high-risk actors
Let $\mathcal{V}'$ be a set of medium-risk actors
Let $\mathcal{V}''$ be a set of low-risk actors
Input: Actor addresses in a Blockchain
Output: $\mathcal{V}, \mathcal{V}', \mathcal{V}''$

1: **function** VERIFY
2: $\mathcal{V}, \mathcal{V}', \mathcal{V}'' \leftarrow \emptyset$;
3: **if** $\mathcal{P}' \not\subseteq P'$ **then**
4: $\mathcal{V} \leftarrow \mathcal{V} \cup \{act \mid \langle act, \alpha, d\rangle \in (\mathcal{P}' \setminus P') \wedge \langle act, \alpha, d, \bot\rangle \in P\}$
5: $\mathcal{V}' \leftarrow \mathcal{V}' \cup \{act \mid \langle act, \alpha, d\rangle \in (\mathcal{P}' \setminus P') \wedge \langle act, \alpha, d, \top\rangle \in P\}$
6: **else**
7: $\mathcal{V}'' \leftarrow \mathcal{V}'' \cup \{act \mid \langle act, \alpha, d\rangle \in \mathcal{P}' \wedge \langle act, \alpha, d, \bot\rangle \in P\}$
8: **return** $\mathcal{V}, \mathcal{V}', \mathcal{V}''$;

Definition 6. Let $\alpha, \alpha' \in \mathcal{A}$ be two operations that have the same type, and an actor *act* executes α prior to α' ($\alpha \preceq \alpha'$) in sequence, as part of a service execution. Moreover, let $d \subseteq \mathcal{D}$ and $d' \subseteq \mathcal{D}$ be the set of personal data requested by α and α', respectively. The operation α' is defined as *data-neutral* if $d' \subseteq d$.

When an operation is data-neutral, details of personal data used do not need to be stored in a Blockchain. This can lead to a reduction in the verification costs (see Section 7.3).

4.3 Contract Activator

A Contract Activator is the coordinator responsible for deploying smart contracts, and identifies the deployment addresses of the data subject, actors involved in processing the data, and the verifier.

5 Architecture: Compliance Checking Phases

In our architecture, GDPR compliance checking is divided into three phases: ratification, submission and verification. The overall coordination is undertaken via the Contract Activator.

5.1 Ratification Phase

This phase requires a data subject to accept/ reject the main purpose of data processing. The sequence diagram in Fig. 6

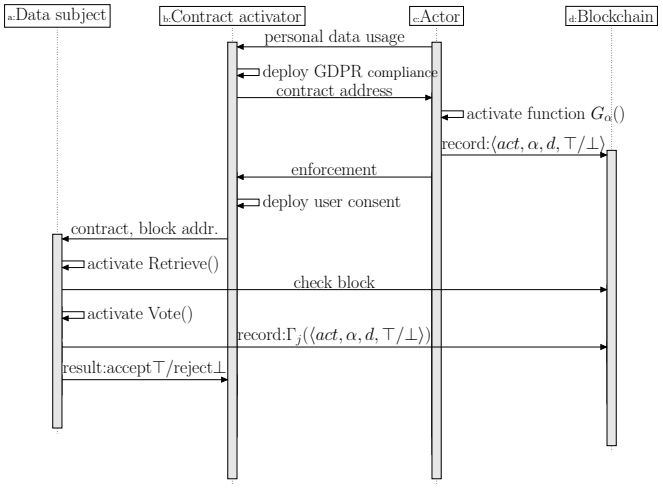

Fig. 6. Interactions in ratification phase.

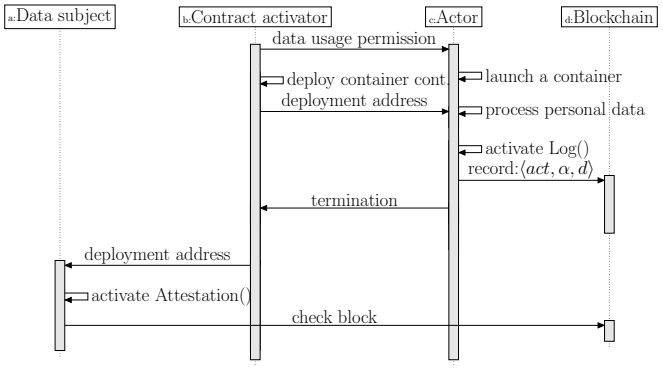

Fig. 7. Interactions in submission phase.

shows the interactions for reaching an agreement between a data subject and an actor. First, an actor sends a message to a Contract Activator for accessing personal user data. The Contract Activator deploys GDPR compliance contract and gives the address of contract deployment to the actor. Using this address, an actor can access the contract and execute function (G_α) based on its purpose of data processing. Once the function is activated, the record forming purpose of data processing ($\langle act, \alpha, d, \top/\bot \rangle$) is submitted to a Blockchain. The actor then sends a message to Contract Activator to get the vote of a data subject ($\Gamma_j(\langle act, \alpha, d, \top/\bot \rangle)$) for the data processing purpose. The Contract Activator deploys a user consent contract and sends the deployment address to a data subject. This address can then be used by the data subject and verified using the Blockchain. The vote of a data subject stored in the Blockchain is accessible for the verification phase. Data subject also notifies Contract Activator about the vote result (accept $\top$/reject $\bot$).

5.2 Submission Phase

This phase records all the information related to the data usage relation set. The interaction between the Contract Activator and an actor for storing such records in a Blockchain is represented by the sequence diagram in Fig. 7. When a data subject confirms the execution of an operation in the previous phase, the Contract Activator sends an acceptance message

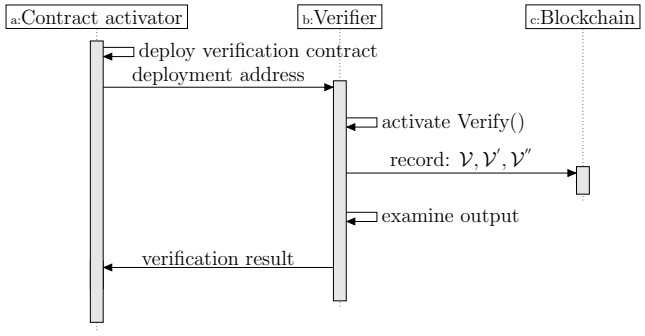

Fig. 8. Interactions in verification phase.

to the actor to access or process personal data. A container is launched on the actor to track operations carried out by the actor on user data. The Contract Activator also deploys a container contract and sends the contract deployment address to the actor – followed by an activation of the *Log* function to record $\langle act, \alpha, d \rangle$. Logging stores the actor address, executed operation and the processed personal data – and subsequently forwarded to a Blockchain. The actor then submits a termination message to the Contract Activator to notify the finalization of data processing. The deployment address of the container contract is then sent to the data subject to individually track activities of actors (achieved through the *Attestation* function supplied by the container contract).

5.3 Verification Phase

This phase verifies data submitted to the Blockchain in previous phases. It detects any violation in GDPR rule(s) that forbids actors to process personal data without the consent of a data subject (Art. 6(1)(a)). The sequence diagram illustrated in Fig. 8 shows the protocol used for this phase. First, the Contract Activator deploys the verification contract and provides the verifier with the address of contract deployment. The verifier then executes the *Verify* function (Algorithm 1) to retrieve blocks created in previous phases. The outputs of the function are addresses of actors violating GDPR rules or executing non-compliant operations ($\mathcal{V}, \mathcal{V}', \mathcal{V}''$) and recorded in a Blockchain for future reference. Finally, the verifier notifies the contract activator of any actors found to violating GDPR rules.

6 Use Case: Cloud-based Pharmacy

Consider an online pharmacy service hosted at a cloud data center, such as one offered by *dincloud* [13]. The service receives patient prescription requests, checks the availability of medicines, and prepares an invoice for the patient. The service also maintains an electronic health record (EHR) system to store useful information about the medical status of patients. The provider of pharmacy service has two subcontractors, a payment and a shipping service provider, in order to handle the payment and delivery of medicines. Figure 9 illustrates an example of a typical cloud-based pharmacy business process comprising of the providers' activities. The payment service provider offers two optional services: western union and paypal to manage the payment process. In case the transfer of money is successful, a receipt is issued and sent to the patient.

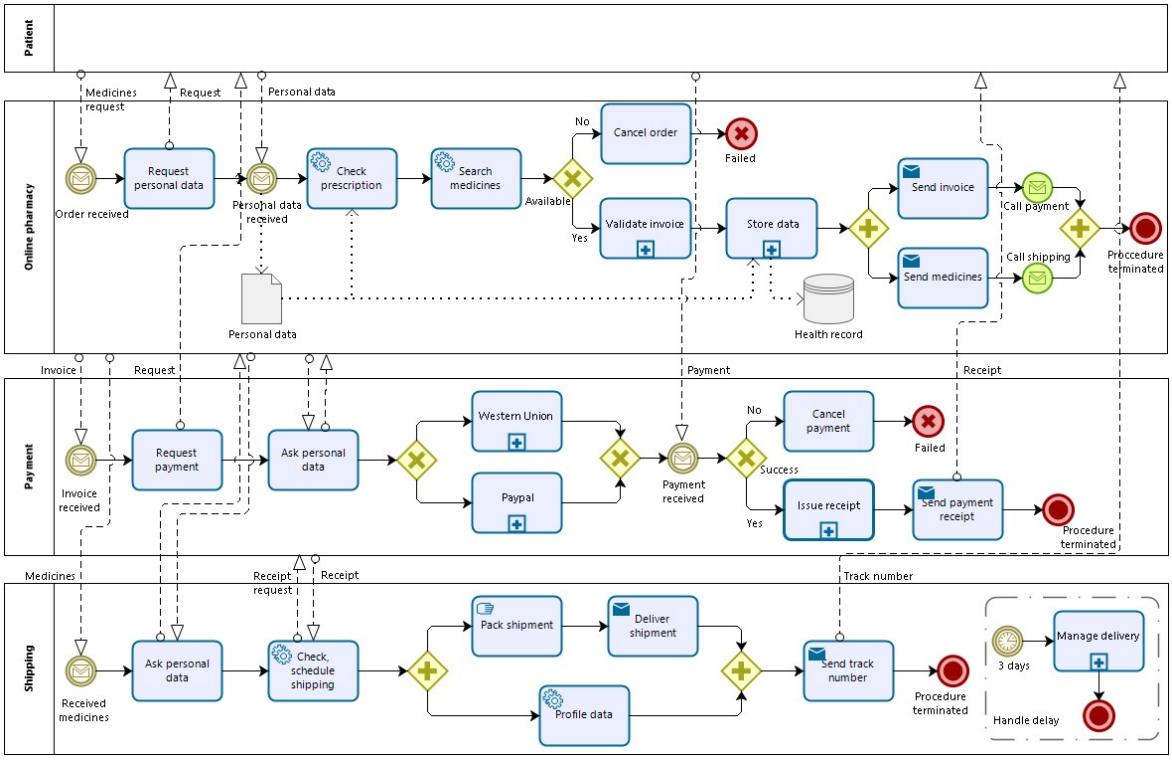

Fig. 9. Business process of online pharmacy and its subcontractors.

The shipping service provider carries out, in parallel, the processes of packing and delivery of medicines and the automated profiling of customer data. It also performs analysis on its customers' personal data to publish several statistical results about the number of parcels sent to a specific region during a period of time. Such information can be used to improve delivery services and to reduce the time to fulfill an order. Finally, the provider sends the customer a reference number for tracking their parcel. There is a constraint in the shipping business process emphasizing the minimal time for delivery of packets (within three days). Each provider requires patients to provide personal data to offer its service. The purposes of data processing for each provider is described below.

Pharmacy service provider requires the following patient data: name, address details, age, general practitioner (GP) diagnosis, electronic version of prescription, and bank account details. It provides the payment service provider personal data including name and bank account details. It also sends the name, age and address details of the patient to the shipping service provider to deliver medicines. The provider maintains the medical information of patients to provide a comprehensive understanding of patients' records for healthcare professionals.

Payment service provider needs the following data: bank account details, provided by the pharmacy service provider to organize the payment process and to transfer money.

Shipping service provider receives the personal data provided by the pharmacy service provider to manage the shipping of medicines. Furthermore, it runs a profiling operation on the destination addresses of its customers to obtain and publish statistics (such as time taken to deliver to a particular address, number of successful/ unsuccessful requests, etc).

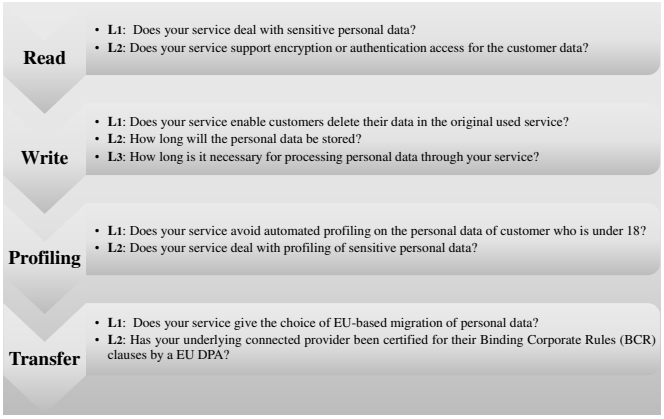

Fig. 10. GDPR legal questions related to each operation.

Regarding the roles defined in GDPR, both payment and shipping service providers are expected to be *data processors* and directly handle or process personal data. The pharmacy service provider, however, can have the roles of *data processor* and *data controller*. It plays the role of a processor when personal data is used for managing and generating the prescription. It plays the role of a controller when personal data is transferred to subcontractors (i.e. other providers).

The operations of these providers on personal data are expressed by typical operations: *read*, *write*, *profiling*, and *transfer*. Specifically, the operations of pharmacy, payment and shipping providers are: (*read, write, transfer*), *read* and (*read, profiling*) respectively.

Assuming that act_1, act_2, and act_3 are pharmacy, payment, and shipping service providers, respectively. The data

usage model of the online pharmacy service with respect to Def. 2 is described as follows.

$$\mathcal{ACT} = \{act_1, act_2, act_3\}$$
$$\mathcal{A} = \{\textit{read, transfer, profiling, store}\}$$
$$\mathcal{D} = \{\textit{name, age, address, GP diagnosis, prescription, bank account}\}$$
$$\mathcal{P} = \{\cdots, \langle act_3, \textit{read}, \{\textit{name, age, address}\}\rangle, \langle act_3, \textit{profiling}, \{\textit{name, age, address}\}\rangle, \cdots\}$$

Set $\mathcal{P}$ is used for the shipping service to provide an illustrative example and to simplify the description of this set. The actual set can contain a number of additional activities. Given the aforementioned operations, Fig. 10 demonstrates the GDPR legal questions assigned to each of them.

Read: Art. 32(1)(a) of GDPR requires actors who read or access sensitive personal data to have an encryption or authentication control mechanism for preventing unauthorized access to data. Two legal questions can be associated with this operation: (i) whether personal data is sensitive or not; (ii) types of mechanisms used for the protection of sensitive personal data [23]. For instance, the GP diagnosis and prescription that are requested by pharmacy service providers can be considered as sensitive data.

Write: Art. 17 of GDPR requires actors who write or store personal data to have a capability for their customers to erase their personal data at anytime. Moreover, Art. 5(1)(e) of GDPR does not allow actors to store personal data longer than the time necessary for data processing. The first legal question is related to Art. 17. The two last questions are based on Art. 5(1)(e).

Profiling: Art. 22 of GDPR states that any automated profiling operation on customers under 18, or whose personal data are in the category of sensitive data, needs to be pre-confirmed. The first legal question relates to analysis operations (e.g. prediction/ forecasting) on personal data of customers who are underage. The second asks actors whether their service deals with profiling of sensitive data or not.

Transfer: Art. 44–47 of GDPR restricts the transfer of personal data outside Europe or outside countries holding Binding Corporate Rules (BCR) certifications [23]. The first and second legal questions are, respectively, related to the geographical location of actors receiving personal data and the BCR status of the data receiver.

6.1 Ratification Phase Business Process

Figure 11 represents the business process in which an actor identifies operations to be executed on personal data, and the data subject gives positive or negative consent. For each type of operation selected by an actor in the GDPR compliance contract, the status of GDPR compliance is automatically checked with respect to the information (inputs) provided by the actor. This status along with the required personal data – referred to as *output* – are subsequently sent to a Blockchain network. In the user consent contract, a data subject can retrieve blocks containing the data (*output*) recorded by the GDPR compliance contract and provide their vote.

In the business process of the cloud-based pharmacy, the pattern (Fig. 11) can be added as a sub-process just before requesting personal data, at the beginning of the process in all service providers.

6.1.1 Verification Operations

The verification of the aforementioned GDPR rules over operations is performed in this phase. As depicted in Fig. 11, the GDPR compliance contract defines a function for each operation which is activated by an actor. Assuming that *compliance* is a Boolean variable, its value shows whether the execution of an operation will comply with its designated GDPR rules or not. For each type of operation, its function outputs are: the value of GDPR compliance status (*compliance*) together with the actor address and the processed personal data. The outputs are recorded in a Blockchain to support verification of actors in the next phases.

G_{read}(): Let $act \in \mathcal{ACT}$ and $d \subseteq \mathcal{D}$ be, respectively, actor address and the set of personal data that must be processed by the actor. Moreover, let *encrypt* be the GDPR-concern element of the read operation. The value of *encrypt* is Boolean and shows whether the service provided by the actor supports encryption of user data or not (declared by *encrypt* is "`true`").

Algorithm 2 Read operation

Input: *act*, *d*, *encrypt*
Output: *act*, *d*, *compliance*

```
1: function G_read
2:     compliance = true;
3:     if encrypt == false then
4:         compliance = false;
5:     return(act, d, compliance);
```

From Algorithm 2, if the value of *encrypt* is "`false`", the operation violates the GDPR rule legislated in Art. 32(1)(a). In the cloud-based pharmacy, all service providers can activate the function, since each requires access to personal data of a user.[2]

G_{write}(): Let $act \in \mathcal{ACT}$ and $d \subseteq \mathcal{D}$ be, respectively, actor address and required personal data. Moreover, let $\mathcal{T}_t$, $\mathcal{T}_s$, and *erase* be the GDPR-concern elements of the operation. The value of $\mathcal{T}_t$ shows the total time taken for data processing and the value of $\mathcal{T}_s$ represents the period of time over which personal data will be stored at the actor. The Boolean variable *erase* determines whether an actor enables customers to erase their data at any time. For example, if the value is "`true`" then the actor has an option for customers to delete their data from the local storage of the actor.

Algorithm 3 Write operation

Input: *act*, *d*, *erase*, $\mathcal{T}_t$, $\mathcal{T}_s$
Output: *act*, *d*, *compliance*

```
1: function G_write
2:     compliance = true;
3:     if erase == false or T_t < T_s then
4:         compliance = false;
5:     return(act, d, compliance);
```

As outlined in Algorithm 3, the execution of *write* operation complies with GDPR if *erase* is "`true`" and $\mathcal{T}_s \leq \mathcal{T}_t$. In the cloud-based pharmacy, the function is activated by the online pharmacy service provider as the provider stores patient records.

2. The function also gets actor address and processed data, since they should directly be sent into a Blockchain to specify the purpose of data processing.

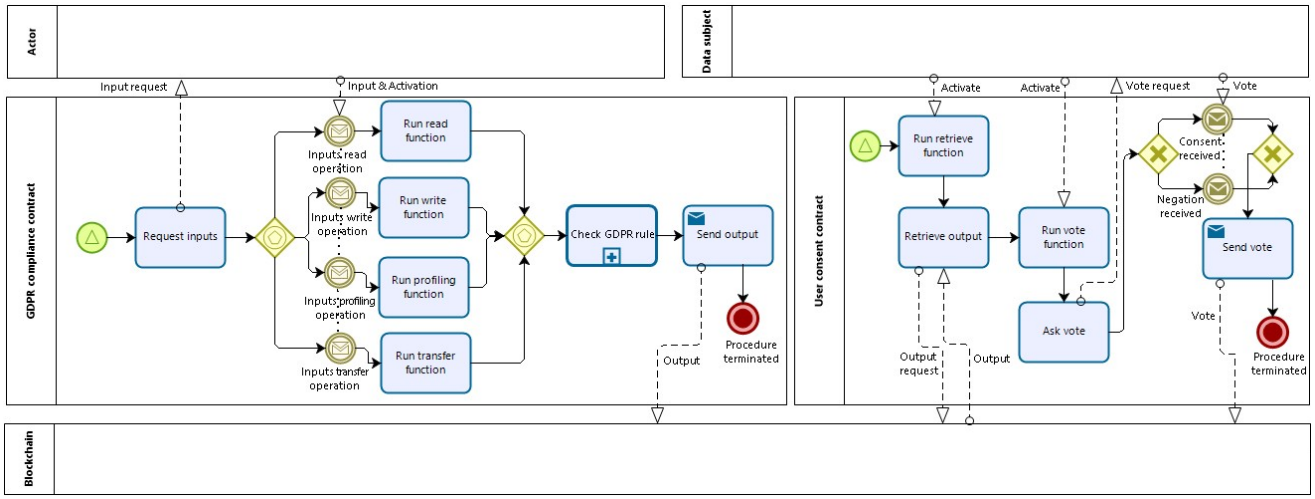

Fig. 11. Business process of ratification phase.

$G_{profiling}()$**:** Let $act \in \mathcal{ACT}$ and $d \subseteq \mathcal{D}$ be, respectively, actor address and the personal data that must be processed. Moreover, let *sensitive* and *isadult* be the GDPR-concern elements of the operation that can have Boolean values. The value of *sensitive* shows whether sensitive data are profiled. The value of *isadult* determines whether the actor performs only profiling operation on adult customers or not (e.g. its "`true`" value denotes the data profiling of adults).

Algorithm 4 Profiling operation

Input: act, d, $isadult$, $sensitive$
Output: act, d, $compliance$

1: **function** $G_{profiling}$
2: $compliance =$ `true`;
3: **if** $isadult ==$ `false` or $sensitive ==$ `true` **then**
4: $compliance =$ `false`;
5: **return**$(act, d, compliance)$;

The profile operation in Algorithm 4 violates GDPR rule (Art. 22) if *sensitive* is "`true`" or *isadult* is "`false`". In the cloud-based pharmacy scenario, the function is only activated by the shipping service provider.

$G_{transfer}()$**:** Let $act \in \mathcal{ACT}$ and $d \subseteq \mathcal{D}$ be, respectively, actor address and the personal data that must be transferred. Moreover, let *loc* be the GDPR-concern element of the operation containing the country name of the data receiver. Assuming that BCR is a set containing the list of countries holding BCR certification, and EU is a set involving the names of European countries.

Algorithm 5 Transfer operation

Input: act, d, loc
Output: act, d, $compliance$

1: **function** $G_{transfer}$
2: $compliance =$ `true`;
3: **if** $loc \notin EU$ **then**
4: **if** $loc \notin BCR$ **then**
5: $compliance =$ `false`;
6: **return**$(act, d, compliance)$;

As seen from Algorithm 5, if personal data is sent outside Europe and the country of data receiver has not been certified by BCR, the value of *compliance* is "`false`". This function should be activated by the online pharmacy service provider, as it transfers personal data to both payment and shipping service providers.

6.2 Submission Phase Business Process

Figure 12 describes a design pattern for the submission phase. When a container is launched, it checks all the activities of the actor on personal data. After the execution of an operation, the address of the actor (act), the name of the operation (α) and the processed personal data (d) are collected and sent to the Blockchain through the container contract. In the online pharmacy business process, the pattern (Fig. 12) can be added as a sub-process after requesting and storing personal data at the actor. In the payment business process, the pattern is added as a sub-process just after asking for personal data. In the shipping business process, the pattern is represented as a sub-process after demanding personal data and also just after analysing/ profiling the data.

6.3 Verification Phase Business Process

Figure 13 shows the business process through which the verification of operations performed by the actors is carried out. Executing the verify function leads to records with user consent, GDPR compliance and container contracts being retrieved from the Blockchain. After checking these records, any violating actors can be reported as described in Algorithm 1. The verifier can give a vote to such a report and the vote is stored in the Blockchain for future reference. Notably, the sub-process *handle delay* is required to notify the verifier about any violation or breaches before 72 hours – as referred to in Art. 33 of GDPR. Due to our Blockchain-based technique, the duty of notifying violations is, however, delegated to the verifier instead of other actors involved in service provision.

After completing the procedures presented in the cloud-based pharmacy business processors, the pattern depicted in Fig. 13 can be executed through the verifier.

7 Experimental Results

A prototype was developed using the Ganache [29] and Ropsten [30] networks. We implemented our proposed smart

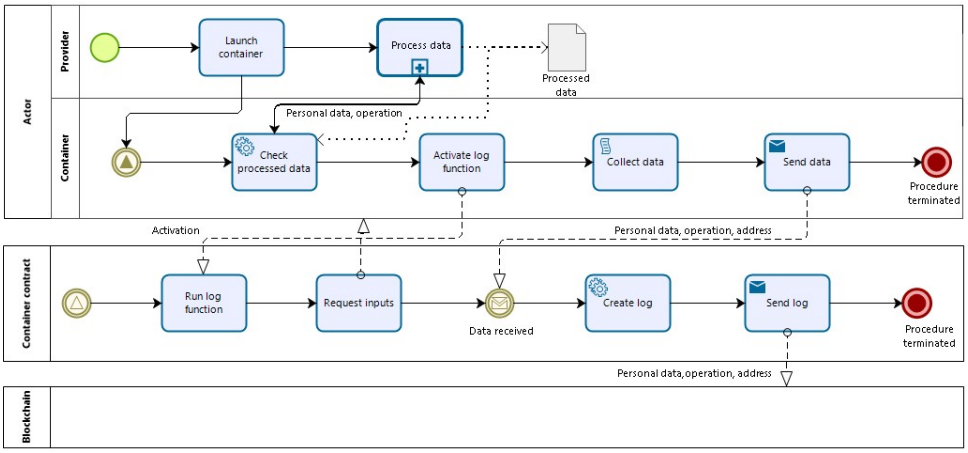

Fig. 12. Business process of submission phase.

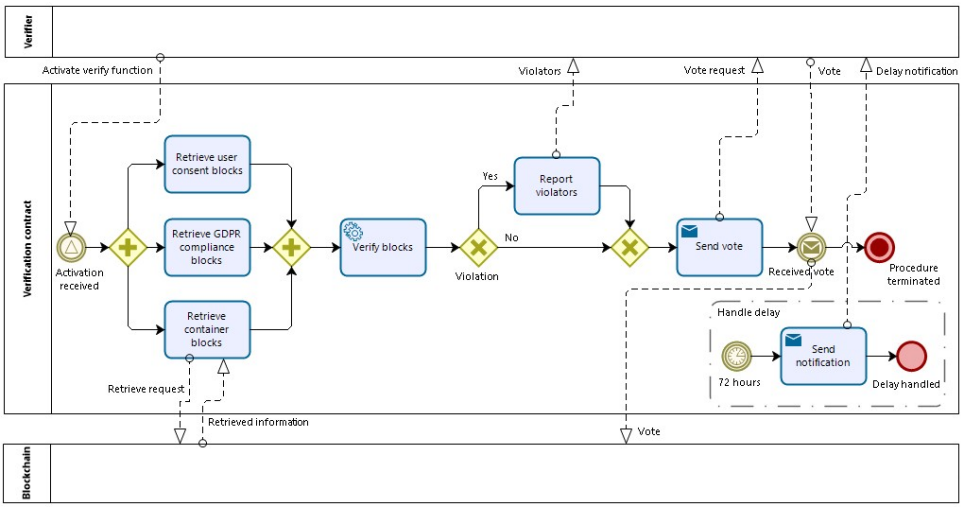

Fig. 13. Business process of verification phase.

contracts on Ethereum using Solidity. The Ganache local test network supplied default gas and ether values that can be used as a currency to alter Blockchain states when executing particular function calls. Ropsten is a public test environment containing a number of miners (and provides detailed information of these miners) and has a gas limit of 4712388 for deploying a contract. The smart contracts were written with a minimum gas consumption for each function execution. They were tested using Remix, which is a Web-based development environment for Solidity running the deployed contracts. The smart contracts *User consent*, *GDPR compliance*, *Container* and *Verification* were deployed in both the Ganache and Ropsten networks. The amount of gas used for contract deployment in both networks was 1156961 for *User consent*, 1494786 for *GDPR compliance*, 527497 for *Container*, and 1246963 for *Verification*. These results show the computational cost of executing each contract. The amount of gas consumed is influenced by the number of actors, and the transaction fees paid for the execution of operations by each actor. The impact of changing the number of actors on the average time taken for the mining process of verification contract is also evaluated. We also investigate the relationship between user payment and GDPR violation detection rate under different scales of operations. The outcome of these experiments can be used to identify the computational capacity needed to support a GDPR-compliance checking system. The more complex the contract, the greater the gas consumption.

7.1 Number of Actors and Gas Consumption

This experiment involves changing the number of actors and evaluates the amount of gas used for the execution of functions in the ratification, submission and verification phases. The number of actors is varied from one, five and ten so that each actor executes an operation randomly selected between *read*, *write*, *profiling* and *transfer*. Moreover, the number of personal data items requested for each operation is randomly changed between one and ten for each execution. Our proposed smart contracts were deployed in the Ganache test network and each function was executed with different parameters, and activated ten times to calculate the average used gas results. Table 1 provides the results of this experiment. When the number of actors (or operations) increases, the amount of consumed gas increases sharply. Since the number of functions or transactions in the ratification phase is more than the other phases, much more should be paid for the execution of this phase compared to the submission and verification phases.

TABLE 1
The relationship between number of actors and used gas

	Actors	Consumed gas	Gas cost (gwei)	Gas cost (USD)
Ratification	1	375190	1125570	$0.3
	5	1482783	4448349	$1.19
	10	2932177	8796531	$2.36
Submission	1	165804	497412	$0.13
	5	729860	2189580	$0.59
	10	1669356	5008068	$1.34
Verification	1	63952	191856	$0.05
	5	282896	848688	$0.23
	10	568885	1706655	$0.46

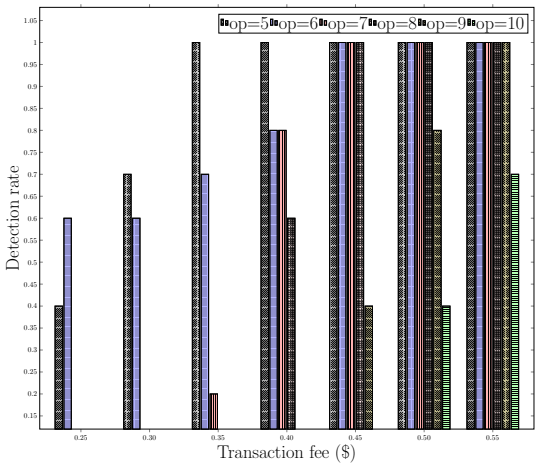

Fig. 14. The relationship between cost and violation detection rate.

The gas price unit is in `gwei`, being 1×10^{-9} ether. The price is 3 `gwei` in the experiment and gas cost is calculated as: *consumed gas* $\times$ *gas price*. As seen from the table, although the calculated gas costs in `gwei` is high, its translation to `USD` is very low.

7.2 Transaction Fee and Violation Detection Rate

This evaluation involves changing the number of operations used, and investigates the relationship between the cost paid by a user for verifying operations and the number of violations detected. The assumption is that we have only one actor and the number of operations executed by an actor varies from five to ten. The number of personal data items required for each operation is randomly selected between one and ten. A gas price of 3 `gwei` is assumed in the experiment. We used the Ganache test network for deploying smart contracts and activating functions in these contracts. Figure 14 demonstrates the test results – with the x-axis showing the cost paid by a user for identifying a violating actor in the verification phase. The y-axis indicates the rate of successful violation detection. Given a specific number of operations and a cost paid for verifying them, the verify function was activated ten times. The number of times that the function was successfully executed is calculated as a detection rate. The experiment

TABLE 2
The relationship between number of data-neutral operations and cost

k	Consumed gas	Gas cost (gwei)	Gas cost (USD)
0	661145	2644582	$0.82
1	655639	2622558	$0.81
2	650142	2600569	$0.80
3	644620	2578481	$0.79
4	639150	2556601	$0.79
5	633655	2534621	$0.78

shows that when the number of operations is five and a user can pay $0.35 for the verification of operations, the verify function is successfully activated even if the operations deal with a maximum number of personal data items (ten in this instance). As illustrated in the figure, there is a direct relationship between the fee paid by a data subject and the rate of violation detection. Moreover, for a given price, when the number of operations increases, the violation detection rate decreases. For example, a violation cannot be detected when the number of operations is ten and our budget is $0.45.

7.3 Effect of Data-Neutral Operations on Verification Fee

This experiment evaluates the impact of the number of data-neutral operations on the gas consumed for the verification. We assume that we have one actor executing ten operations on user data. Each operation requests a number of personal data items, selected randomly between one and ten. We used Ganache for deploying smart contracts and executing their functions. The gas price was 4 `gwei` in the experiment. Table 2 represents the details of test results, where the values are approximate. The number of data-neutral operations k varies from 0 to 5. We calculated the average gas used by the verify function after five executions of smart contracts. As observed from the table, when the number of data-neutral operations increases, the gas consumption of the function decreases slightly. For instance, a comparison between $k = 0$ and $k = 5$ shows that the cost of the verify function has a difference of $0.04.

7.4 Number of Actors and Verification Mining Time

This experiment evaluates the relationship between the number of actors and the mining time taken for their verification. The number of actors varies from one to ten and each executes an operation randomly chosen amongst the ones in the case study. Moreover, the number of personal data asked for is varied between one and ten per execution. We used the Ropsten test network to get the results of this experiment, as Ropsten can be used to calculate the time (in seconds) between the activation to the mining of a transaction. Figure 15 illustrates the time for the verify function to be successfully mined, from its initial activation time. In fact, the smart contracts were executed five times to calculate the average mining time of the verify function. As seen from the figure, this time fluctuates when we change the number of actors, and is (primarily) influenced by the interest of miners in executing the verify function. It does not depend on the number of actors or the

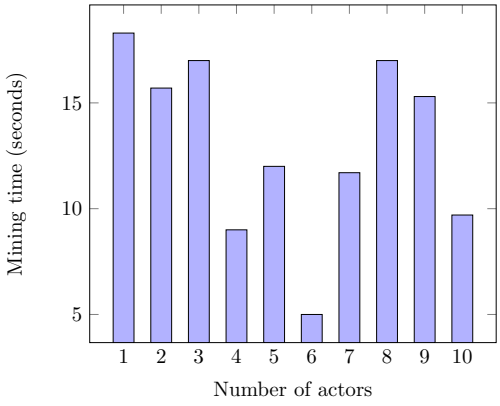

Fig. 15. The relationship between mining time and actors.

function parameters. As a result, the miners can usually take an arbitrary time for the mining process.

8 Related Work

The challenge of user trust and privacy for sharing data in a cloud ecosystem has recently motivated cloud researchers to find a solution based on smart contracts and Blockchain-based techniques. A Blockchain-based data sharing framework was presented to provide privacy for recording medical data within cloud environments [37]. The framework was based on a lightweight and permissioned Blockchain giving access rights to only verified users. The potential of using Blockchain-based techniques in order to protect healthcare data located in cloud was extensively studied in [32]. The authors described practical challenges to highlight the importance of privacy in recording medical data in a Blockchain network. The authors in [33] presented a patient centric healthcare data management system with the aid of a Blockchain network. The system ensures that private healthcare data in a cloud system is only accessible by a patient. An approach for building a public ledger for supporting policy compliance was proposed in [34]. The approach introduced an off-chain channel to provide the verification of external parties to the information stored in a Blockchain. In [11], a Blockchain-based approach was proposed for storing cloud attestation. The authors implemented a smart contract for recording the migration of user data between cloud providers. The deployment of the smart contract enabled cloud users to be informed about the location of their data through the submission of a query to the contract. In [39], the same authors extended a smart contract implemented in [11] to provide cloud users more control on the migration of their data so that data can only be shared between providers existing in a users' white list. In [39], the trust of cloud customers was improved by improving visibility of data movement policies for users, which was realized using a Blockchain-based technique [35]. The authors in [38] proposed an automatic way for tracking and enforcing data sharing agreements between a user and cloud providers with the aid of smart contracts and a Blockchain network. In this approach, the providers who violated the shared agreements were detected through a set of voters or arbiters listed in a voting contract. In [36], a secure smart home architecture based on cloud and Blockchain technology was proposed. The authors used an encryption and hashing algorithm in Blockchain technology to obtain confidentiality and trust in smart home networks. The integration of Blockchain-based approaches with several security services, including authentication, privacy, data provenance and integrity has been reviewed in [9]. Given some recent approaches in the data provenance domain, a conceptual model—called *ProvChain*—was designed to collect cloud data provenance and provide assurance for data operations in a cloud. This was achieved by analysing the provenance log stored in a Blockchain network [40]. Although the aforementioned approaches take advantage of Blockchain and smart contracts to enhance cloud user privacy and trust, none of them focus on GDPR rule violation. There is also limited support provided in existing approaches for mapping data privacy legislation into automated rules that can be verified through cloud provider monitoring logs.

Looking at recent contributions that benefit from combining both Blockchain and GDPR, in [17], a Blockchain-based approach for supporting data accountability and provenance tracking, which meets GDPR requirements, was proposed. The approach presented two different models for deploying a smart contract in a Blockchain network. The first model used data subject consent rules in a Blockchain under which each actor (controller/ processor) should follow the rules. The second model deployed actor policies as a smart contract in a Blockchain that allows users as subscribers to join or leave the contract. However, the verification of Blockchain to check whether actors violated consent rules was manually undertaken by a user in both models. Moreover, a combination model to enable negotiation between a data subject and an actor for reaching a shared agreement was not studied in [17]. A personal health data sharing system has been proposed in [41], which enabled users to securely share their health data and help data consumers to get necessary data in a transparent manner and in compliance with GDPR. The system used Blockchain technology supplemented by cloud storage to share the health data. Likewise, a data quality inspection module relied on machine learning approaches was introduced in the system to monitor the quality of personal health data. Although the system benefits from GDPR and Blockchain for improving the privacy of users' health data, it still lacks a methodology whereby the verification of stored data in the Blockchain network is supported. The authors in [19] designed a conceptual and a high level architecture for an identity management system that provides control on personal data usage with the aid of GDPR. The architecture also utilized Blockchain technology to supply transparency, trust, and security. However, the validation and deployment of architecture in real-world applications was not discussed. A Blockchain-based personally identifiable information management system, called BcPIIMS, was proposed in [20] so that storing personal data in the system complied with a GDPR rule. Though, the management system has not yet implemented in practice. Furthermore, the verification of GDPR rules on the system was only limited to the rule: *right to be forgotten*. The authors in [21] took advantages of Blockchain and GDPR to develop a digital onboarding framework that defines some security policies for users' identity attributes stored on multiple centralized repositories. However, the automatic verification of GDPR rules over the framework was not studied. In [23], a privacy-aware cloud architecture making use of GDPR and Blockchain was de-

signed to enhance transparency and enable the audit trail of providers who accessed user data. Though, the GDPR rules legislated for user consent and data storage were not examined and the notions of data usage and purpose of data processing were not formally defined. Moreover, the validation of the architecture and the transaction costs of its smart contracts were not investigated. In [42], a semantic model was proposed through which Blockchain and smart contract were used to check a data subject's consent over the activities of data processors. GDPR compliance checking of the model, however, was only limited to consent requirements of a data subject, and integration of the model in cloud environment was not discussed. The authors in [43] analysed GDPR concerns that should be considered (in various applications) using a Blockchain network to track activities of data processing actors on personal data. A practical solution for the automatic verification of GDPR rules, however, was not examined in this approach. A GDPR-compliant data management platform using a Blockchain network was proposed in [44]. The platform provided a decentralised mechanism for cloud providers and their customers to process personal data and improve data provenance with the aid of a Blockchain. However, the platform only verified GDPR rules related to user consent. The cost of data storage and computational requirements used in the verification process were not evaluated in this work.

9 Conclusion

This paper enhanced the privacy of cloud users through the integration of Blockchain and GDPR rules. It provided a foundation for translating GDPR rules into smart contracts that enabled the verification of cloud providers in an automatic and transparent way. It formally defined the types of operations that can be executed on user data, and the purposes of data processing by providers. This was used to determine what information should be recorded in the Blockchain network to facilitate subsequent GDPR-compliance verification. A system and functional architecture was designed to show how cloud users and providers can connect to a Blockchain network and access smart contracts that can be used for complying with GDPR. The architecture enabled users to provide positive (or negative) consent for executing provider operations, and for recording these operations (and the associated consent outcomes) in a Blockchain network for futher analysis. Furthermore, the architecture introduced a Blockchain-based technique that supports both preventative and reactive mechanisms to support GDPR compliance checking. The former allowed only providers to operate on personal data if their purposes of data processing were already accepted by user. The latter verified the operations appearing in the purposes of data processing after their execution on personal data and reported any provider activity that conflicts with user consent. An illustrative case study was also presented to demonstrate how the proposed approach can be used. This involves a cloud-based pharmacy service that supports four typical operations carried out by providers on customer data. For each operation, several GDPR related legal questions (and related to particular clauses in this regulation) were assigned to enforce providers to give summarized information about their services, dealing with personal data. Moreover, some algorithms were proposed in the case study for verifying operations with respect to GDPR rules. The case study also showed how our Blockchain-based approach in forms of sub business processes can be placed in the main design pattern of cloud-based pharmacy. Finally, our proposed smart contracts were deployed in both local and global Blockchain test networks (Ganache and Ropsten) and experiments were conducted to indicate how much gas is consumed to check GDPR compliance of provider activities on personal data.

Future work will focus on translating other available GDPR rules, particularly those legislated for cloud [10], into smart contracts. Particularly, we will investigate how the GDPR rule "the right to be forgotten" can be supported by our proposed technique, since the rule currently has a conflict with the immutability feature of a Blockchain network. The implementation of the designed architecture in a permissioned or private Blockchain such as Hyperledger Fabric is also another potential research avenue for future consideration.

Acknowledgement: This work has been carried out in the "PACE: Privacy-aware Cloud Ecosystems" project, funded by EPSRC under project grant: EP/R033439/1.

References

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, An overview of Blockchain technology: Architecture, consensus, and future trends, *in Proc. of the IEEE 6th International Congress on Big Data*, Honolulu, USA, 2017, pp. 557–564.

[2] "Bitcoin GitHub Implementation." Accessed: March 2020. [Online]. Available: https://github.com/bitcoin/bitcoin

[3] "Corda." https://www.corda.net/, March 2020. [Online].

[4] "R3." https://www.r3.com/, March 2020. [Online].

[5] "Monax." https://monax.io/, March 2020. [Online].

[6] "Multichain." https://www.multichain.com/, March 2020. [Online].

[7] "ETH Gas Station." https://ethgasstation.info/, March 2020. [Online].

[8] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, *Ethereum Project Yellow Paper*, 2014.

[9] T. Salman, M. Zolanvar, A. Erbad, R. Jain, and M. Samaka, Security services using Blockchains: A state of the art survey, *IEEE Communications Surveys and Tutorials*, 2018.

[10] M. Corrales, P. Jurcys and G. Kousiouris, Smart contracts and smart disclosure: Coding a GDPR compliance framework, *SSRN Electronic Journal*, 2018.

[11] S. Kirkman and R. Newman, Using smart contracts and Blockchains to support consumer trust across distributed clouds, *in Proc. of the 13th International Conference on Grid, Cloud, and Cluster Computing*, Las Vegas, NV, 2017, pp. 10–16.

[12] "Ethereum." https://www.ethereum.org/, March 2020. [Online].

[13] "dincloud." https://www.dincloud.com/cloud-for-pharmacy, March 2020. [Online].

[14] K. Bila, O. Khali, A. Erbad, and S. U. Kha, Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers, *Computer Networks*, vol. 130, pp. 94–120, 2018.

[15] B. Russo, L. Valle, G. Bonzagni, D. Locatello, M. Pancaldi, and D. Tosi, Cloud computing and the new EU general data protection regulation, *IEEE Cloud Computing*, vol. 5, no. 6, pp. 58–68, 2018.

[16] M. Virvou and E. Mougiakou, Based on GDPR privacy in UML: Case of e-learning program, *in Proc. of the 8th International Conference on Information, Intelligence, Systems & Applications*, Larnaca, Cyprus, 2017.

[17] R. Neisse, G. Steri, and I. Nai-Fovino, A Blockchain-based approach for data accountability and provenance tracking, *in Proc. of the 12th International Conference on Availability, Reliability and Security*, Reggio Calabria, Italy, 2017.

[18] D. Ulybyshev, M. Villarreal-Vasquez, B. Bhargava, G. Mani, S. Seaberg, P. Conoval, R. Pike, and J. Kobes, (WIP) Blockhub: Blockchain-based software development system for untrusted environments, *in Proc. of the 11th International Conference on Cloud Computing*, San Francisco, CA, 2018, pp. 582–585.

[19] B. Faber, G. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, BPDIMS: A Blockchain-based personal data and identity management system, *in Proc. of the 52nd Hawaii International Conference on System Sciences*, Hawaii, USA, 2019, pp. 6855–6864.

[20] N. Al-Zaben, M. M. H. Onik, J. Yang, N-Y. Lee, and C-S. Kim, General data protection regulation complied Blockchain architecture for personally identifiable information management, *in Proc. of the International Conference on Computing, Electronics & Communications Engineering*, Southend, UK, 2018, pp. 77–82.

[21] R. Soltani, U. T. Nguyen, and A. An, A new approach to client onboarding using self-sovereign identity and distributed ledger, *in Proc. of the IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics*, Halifax, Canada, 2018, pp. 1129–1136.

[22] G. Zyskind, O. Nathan, and A. S. Pentland, Decentralizing privacy: Using Blockchain to protect personal data, *in Proc. of the IEEE Security and Privacy Workshop*, CA, USA, 2015, pp. 180–184.

[23] M. Barati, O. Rana, G. Theodorakopoulos, and P. Burnap, Privacy-aware cloud ecosystems and GDPR compliance, *in Proc. of the 7th International Conference on Future Internet of Things and Cloud*, Istanbul, Turkey, 2019, pp. 117–124.

[24] T. Al Said, O. F. Rana, and P. Burnap, VMInformant: An instrumented virtual machine to support trustworthy cloud computing, *International Journal of High Performance Computing & Networking (IJHPCN)*, vol. 8, no. 3, pp. 222–234, 2015.

[25] S. Hosseinzadeh, S. Laurén, and V. Leppänen, Security in container-based virtualization through vTPM, *in Proc. of the 9th International Conference on Utility Cloud Computing*, Shanghai, China, 2016, pp. 214–219.

[26] V. Costan and S. Devadas, Intel SGX explained, *Cryptol. ePrint Arch., Tech. Rep. 2016/086*, 2016. [Online]. Available: https://eprint.iacr.org/2016/086

[27] E. Bacis, S. Mutti, S. Capelli, and S. Paraboschi, DockerPolicyModules: Mandatory access control for docker containers, *in Proc. of the International Conference on Communications and Network Security*, Florence, Italy, 2015, pp. 749–750.

[28] Y. Sun, D. Safford, M. Zohar, D. Pendarakis, Z. Gu, and T. Jaeger, Security namespace: making linux security frameworks available to containers, *in Proc. of the 27th USENIX Security Symposium*, Baltimore, USA, 2018, pp. 1423–1439.

[29] "Ganache." https://github.com/trufflesuite/ganache, March 2020. [Online].

[30] "Ropsten testnet pow chain." https://github.com/ethereum/ropsten, March 2020. [Online].

[31] "Solidity." https://solidity.readthedocs.io/en/v0.5.3/, March 2020. [Online].

[32] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K-K R. Choo, Blockchain: A panacea for healthcare cloud-based data security and privacy?, *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018.

[33] A. A. Omar, M. Z. A. Bhuiyan, A. Basuc, S. Kiyomoto, and M. S. Rahman, Privacy-friendly platform for healthcare data in cloud based on Blockchain environment, *Future Generation Computer System*, vol. 95, pp. 511–521, 2019.

[34] Z. Wu, A. B. Williams, and D. Perouli, Dependable public ledger for policy compliance, a Blockchain based approach, *in Proc. of the 39th International Conference on Distributed Computing Systems*, Dallas, TX, 2019, pp. 1891–1900.

[35] S. Kirkman, A data movement policy framework for improving trust in the cloud using smart contracts and Blockchains, *in Proc. of the IEEE International Conference on Cloud Engineering*, Orlando, FL, 2018, pp. 270–273.

[36] S. Singh , I-H. Ra, W. Meng, M. Kaur, and G. H. Cho, SH-BlockCC: A secure and efficient Internet of things smart home architecture based on cloud computing and Blockchain technology, *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, 2019.

[37] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, BBDS: Blockchain-based data sharing for electronic medical records in cloud environments, *Information*, vol. 8, no. 2, p. 44, 2017.

[38] H. Desai, K. Liu, M. Kantarcioglu, and L. Kagal, Enforceable data sharing agreements using smart contracts, *arXiv:1804.10645v1[cs.CY]*, 2018.

[39] S. Kirkman and R. Newman, A cloud data movement policy architecture based on smart contracts and the Ethereum Blockchain, *in Proc. of the IEEE International Conference on Cloud Engineering*, Orlando, FL, 2018, pp. 371–377.

[40] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, ProvChain: A Blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability, *in Proc. of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Madrid, Spain, 2017, pp. 468–477.

[41] X. Zheng, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Mere, Blockchain-based personal health data sharing system using cloud storage, *in Proc. of the 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Ostrava, Czech Republic, 2018.

[42] M. Davari and E. Bertino, Access control model extensions to support data privacy protection based on GDPR, *in Proc. of the Int. Conf. on Big Data*, Los Angeles, CA, 2019, pp. 4017–4024.

[43] F. Zemler and M. Westner, Blockchain and GDPR: Application scenarios and compliance requirements, *in Proc. of the Portland Int. Conf. on Management of Engineering and Technology*, Portland, OR, 2019.

[44] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, GDPR-compliant personal data management: A Blockchain-based solution, *IEEE Trans. on Information Forensics & Security*, vol. 15, pp. 1746–1761, 2020.

Masoud Barati received the PhD degree in computer science from Sherbrooke University, Canada, in 2018. He is currently a Research Associate with the School of Computer Science & Informatics, Cardiff University. His research interests include distributed systems, Blockchain, formal methods, and cybersecurity.

Omer Rana received the PhD degree in "neural computing and parallel architectures" from the Imperial College, University of London. He is a professor of performance engineering in the School of Computer Science & Informatics, Cardiff University & a member of Cardiff University's "Data Innovation Institute". His research interests include distributed systems, Blockchain, and scalable data analysis.